



DTO Java Record #2

Transcripción

[00:00] Entonces vamos a realizar el payload y vemos que aquí tenemos un error. Lo que vamos a hacer es, ahora ya no está apareciendo con error. Tenemos ya ortopedia mayúscula. Vamos a enviarlo ahora y vemos aquí te da un 200, 200 que es éxito. Aquí como puedes ver tienes el dato registro médico con un formato digamos de llave valor.

[00:23] Nombre Rodrigo López, el email Rodrigo.lopez@voll.med y todos los datos que hemos enviado aquí desde nuestro front end, desde nuestro cliente, pero por ejemplo, ¿qué pasaría si yo no le envío calle, no le voy a enviar calle? Vamos a ver qué pasa. Me da 200, igual. Veo aquí y vemos que simplemente en calle si es que no recibe el parámetro a diferencia de en el primer caso ahora imprime null, porque simplemente el objeto string de calle está vacío.

[00:58] No hay nada más que ver por ahora. Tienen que practicar mucho este mapeo de parámetros y el mapeo en los records. Tenemos por ahora 2 DTO. Este tipo de patrón que estamos usando se llama patrón DTO, Data Transfer Object, que es básicamente usar a nivel de controller un objeto como intermediario para que mapee la información que nos llega desde nuestro cliente hacia nuestro API.

[01:30] Por ejemplo, si yo le envío también algo que no está esperando mi API, si le envío, no sé, algo como tipo de sangre, entonces simplemente no va a ser recibido ni procesado aquí porque mi DTO, mi Data Transfer Object, no tiene mapeado ese parámetro. Entonces, este es un patrón muy utilizado.

[01:50] Solo antes de irnos, DatosDirección, voy a hacer exactamente lo mismo que hice con datos med, voy a crear aquí un nuevo paquete llamado dirección, porque también va a ser usado por el paciente, entonces lo voy a mover a ese nuevo paquete, le doy okay.

[02:05] Y tenemos nuestro paquete dirección, nuestro paquete médico, con los DatosRegistroMedico y la especialidad. Tienen alguna duda, pueden dejarla en el foro. Lo que viene a continuación son las validaciones.

[02:18] Si volvemos aquí a nuestras reglas de negocio, bueno, en las consideraciones que debemos tener son que recibamos solo letras, no pueden llegar vacíos, dentro de él tiene que ser valores dentro de enum, aquí solo números. Entonces, tenemos ese tipo de situaciones que las vamos a ver en la siguiente clase. Nos vemos.